Predictably un-predictable — on the implementation of a Walking Pattern Generator for the full-sized humanoid robot HUBO2 using Model Predictive Control

David T. Butterworth[1] *Member, ARAA*
Boon Siew Han[2]
Surya P. N. Singh[1] *Member, ARAA*

***Abstract*— There is a large body of research related to WPG (Walking Pattern Generators) for humanoid robots. Typically WPG are evaluated based on how well the robot's actual ZMP (Zero Moment Point) tracks the desired ZMP trajectory, using a simulation of a rigid-body robot walking on a solid floor. However little has been written about how various approaches scale-up to a full-sized humanoid robot, which has unmodeled compliance in the joints or contact surfaces, and makes contact with non-rigid surfaces in the environment like a soft floor. This paper compares the implementation of three WPG: Parametrized Polynomials, Preview Control and MPC (Model Predictive Control), and shows results from simulation and initial testing on the 1.25m tall humanoid robot HUBO2.**

I. INTRODUCTION

Humanoid robots will be useful for applications involving tools or environments that were designed for humans, such as elder care and disaster response. Their anthropomorphic appearance also makes them appealing for social robotics experiments and promoting education in STEM (Science, Technology, Engineering and Mathematics). The Social Robotics Lab at A*STAR in Singapore has been using a HUBO2 robot to investigate potential future applications for humanoid robots such as office work and home security [1]. But for such a robot to be truly useful within a home or office environment, it must be capable of safe, long-term autonomous operation, which for a biped means reliable walking ability.

To date, most humanoids like ASIMO [2], WABIAN-2 [3], HRP-2 [4], and HUBO2 [5] use a ZMP (Zero Moment Point) [6] based WPG (Walking Pattern Generator) and high-gain position-control. Other approaches have framed walking as part of the push recovery problem [7] or by regulating the Capture Point [8].

The typical approach for ZMP-based walking is to combine a WPG with various stability controllers such that the rigid-body humanoid precisely tracks the desired trajectory. In this paper we compare three methods for generating the walking pattern with respect to the walking stability and compute time for the humanoid robot HUBO2. There is not much information in the literature about how these algorithms scale-up to full-sized robot and so the contribution of this paper is to show how the results in simulation compare to initial testing on the real robot. It was found that using a more complex algorithm does not directly translate to more stable walking and the key issues are discussed.

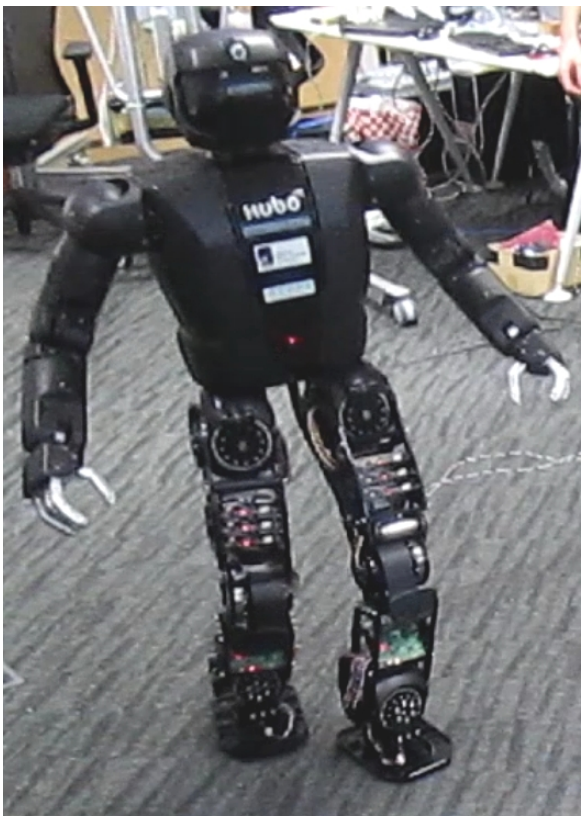

Fig. 1. HUBO2+ walking at the A*STAR Social Robotics Lab

II. METHODS FOR GENERATING WALKING PATTERN

ZMP-based walking relies on assumptions inherent to a position-controlled humanoid, being that the robot has rigid links, high-gain joints and flat feet that make parallel floor contact with no slip. There are numerous approaches to generating walking patterns; Parametrized Cubic Polynomials [9–12], CPG (Central Pattern Generator) [13] and Fourier Series [14] are "top down" methods, where the foot positions are modified to stabilize the COM (Center of Mass); Preview Control [15] and MPC (Model Predictive Control) [16] are

[1]David Butterworth and Surya Singh are with the School of Information Technology and Electrical Engineering, University of Queensland, Australia dbutterworth@uq.edu.au spns@uq.edu.au

[2]Boon Siew Han is with the Institute for Infocomm Research, Agency for Science Technology and Research, Singapore

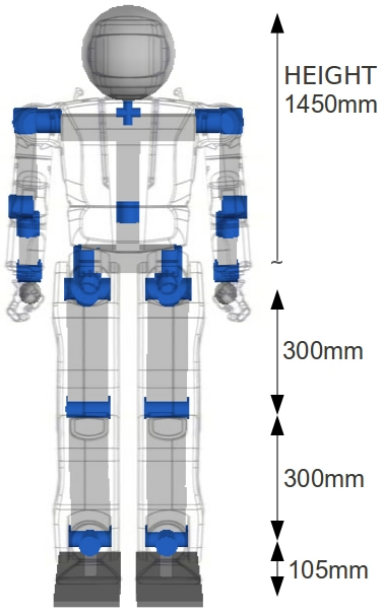

Fig. 2. Simulation model of the HUBO2 robot

"bottom up" approaches where the COM is modified to suit constrained footstep positions. This is a useful distinction because a bottom-up approach allows the biped to nimbly step amongst clutter or on uneven terrain, using the direct output of a footstep planner.

A. PARAMETRIZED POLYNOMIALS

A WPG based on cubic polynomials or sinusoids uses these functions to generate trajectories for the COM (Center of Mass) and feet of the robot. The robot is typically modeled using a LIPM (Linear Inverted Pendulum Model), with the robot's trunk or pelvis assumed to be the position of the COM, and the trajectory parameters adjusted to create a ZMP trajectory that maintains stability. The resulting combination of the x, y trajectories for the trunk and feet of the robot creates the desired walking pattern.

A detailed explanation of applying cubic polynomials to walking patterns can be found in [17]. The method is popular because it has low computational load and so can be applied to bipeds of any capacity. Xue et al. [11] developed a real-time WPG using cubic polynomials that allows the trajectory direction to be changed within a window of 1 footstep, whilst maintaining a smooth ZMP trajectory, and demonstrated it on the NAO robot. Hong et al. [12] used quartic polynomials and Least Squares method to design walking patterns that maintained smooth jerk despite a variable step length.

The HUBO2 robot currently uses a WPG that superimposes linear, sinusoidal and polynomial functions to create smooth, cyclic motions of the pelvis and feet [9, 10]. Its low complexity means it can run on a low-power Blackfin Microcontroller with control loop times ¡ 5ms [18]. The limitations of this method are that the footstep positions are not easily constrained, and that gaits of varying step size or speed must each be designed by hand as the walking pattern does not factor the dynamic stability.

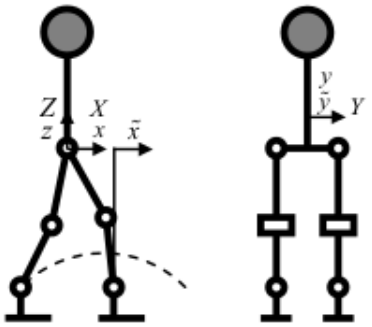

Fig. 3. Coordinate frames for walking: sagittal (x) & coronal (y) [10]

TABLE I
PARAMETERS FOR POLYNOMIAL WPG

Parameter	Description
N_{step}	No. of steps
L_{step}	Step length
H_{step}	Step height
R_{step}	Step rotation (yaw)
D_{step}	Step direction (forward/sideways)
T_{stride}	Step period (speed)
A_{pelvis}	Hip sway amplitude
T_{delay}	Hip sway delay

The walking pattern is designed based on a LIPM (Linear Inverted Pendulum Model) and then experimentally tuned and verified on the real robot. The function parameters are adjusted to create a set of walking movement primitives that allow the robot to walk forwards in increments of $0-20cm$, backwards $0-10cm$, and turn at angles of $\pm 0-30°$.

Trajectories for the feet and pelvis are designed in body-fixed coordinates $x-y-z$ (Fig. 3) and then translated into world coordinates $X-Y-Z$. A FSM (Finite State Machine) is used to generate each phase of the walking pattern $\tilde{x}, \tilde{y}, \tilde{z}, y\tilde{a}w, s\tilde{w}ay$ based on the parameters shown in Table I.

Using the point-mass model, a reference trajectory for the x ZMP is first designed using 3rd order polynomial interpolation (1). The coefficients are derived from the boundary conditions, being the position and velocity at the start time $t_N = 0$ and end time $t_N = 1$ of each step [10].

$$x_{zmp} = \sum_{i=0}^{3} b_i t_N^i \tag{1}$$
$$= (a_0 - 2a_2\frac{l}{g}) + (a_1 - 6a_3\frac{l}{g})t_N + a_2 t_N^2 + a_3 t_N^3$$

The y ZMP trajectory is similar, but the cubic polynomial is defined piece-wise with an additional linear segment centered around $\frac{t}{2}$. This sway delay time T_{delay} holds the y ZMP position constant during the SSP (Single Support Phase), with zero velocity and acceleration.

By assuming that the location of the robot's pelvis center $x-y$ with fixed height z is the location of the COM from the

LIPM, the trajectory of the pelvis will also be a 3rd order polynomial but with different coefficients. The validity of this approach and a full derivation is discussed in [10].

The x trajectory of the pelvis center is generated using the following cubic polynomial:

$$\begin{aligned}\tilde{x}_{pelvis}(t) &= \sum_{i=0}^{3} a_i \left(\frac{t-t_1}{t_2-t_1}\right)^i \\ &= a_3\left(\frac{t-t_1}{t_2-t_1}\right)^3 + a_2\left(\frac{t-t_1}{t_2-t_1}\right)^2 \\ &\quad + a_1\left(\frac{t-t_1}{t_2-t_1}\right) + a_0\end{aligned} \tag{2}$$

where t_1 and t_2 define the time period of the swing foot between each DSP.

The y trajectory of the pelvis center is generated using 2 cubic polynomials and 1 linear segment:

$$\tilde{y}_{pelvis}(t) = \begin{cases} \sum_{i=0}^{3} \bar{a}_i \left(\frac{t-t_1}{t_0-t_1}\right)^i & \text{for } t = t_1 \le t < t_0 \\ -\beta_1 S_y & \text{for } t = t_0 \\ \sum_{i=0}^{3} \bar{a}_i \left(\frac{t-t_0}{t_2-t_0}\right)^i & \text{for } t = t_0 < t \le t_2 \end{cases} \tag{3}$$

noting that $t_0 = \frac{t_1+t_2}{2}$.

The trajectory of the swing ankle position $\tilde{x}$ is described by a cycloid function:

$$\tilde{x}_{ankle}(t) = (b+f)\left(\frac{t-t_1}{t_2-t_1} - \frac{1}{2\pi}\sin\left(2\pi\frac{t-t_1}{t_2-t_1}\right)\right) - b \tag{4}$$

where $b = \tilde{x}_{ankle}$ at $t = t_1$ and $f = \tilde{x}_{ankle}$ at $t = t_2$ are with respect to the frame x of the stance foot.

The y trajectory of the left ankle is generated using 2 cosine segments:

$$\tilde{y}_{ankle_{left}} = \begin{cases} \frac{A}{4}(1-n)\left(1-\cos\left(\pi\frac{t-t_1}{t_2-t_1}\right)\right) & \text{for } t = t_1 \le t < t_0 \end{cases} \tag{5}$$

$$\begin{cases} \frac{A}{4}(1-n)\left(1+\cos\left(\pi\frac{t-t_1}{t_0-t_1}\right)\right) & \text{for } t = t_0 < t \le t_2 \end{cases} \tag{6}$$

where n is the side-to-stride ratio and the 2nd segment is simply the first time-shifted by one half period. The right ankle is similar, but has negative A for the stride length.

This WPG has been used successfully on the real HUBO2 robot, however it does not factor in the dynamics or current state of the robot. The various parameters (Table I) must be tuned to suit each gait primitive and on uneven ground or during sharp turns the stability margin can be so low that the robot must always be operated with fall-arrest protection.

The next section will discuss the ZMP Preview algorithm, which incorporates a dynamic model of the robot.

B. Preview Control

This section will briefly describe Kajita's WPG using Preview Control [15], which is an infinite horizon LQR (Linear Quadratic Regulator) with integral term. In contrast to the top-down approach in the previous section, this method is a bottom-up approach that generates a dynamically-stable trajectory to suit fixed footstep positions. A planning algorithm will output discrete footstep positions which is assumed to be the location of the ZMP in each SSP (Single Support Phase). The ZMP is sampled on a preview horizon to compute the COM, which is assumed to be the location of the waist. Detailed descriptions of the algorithm can be found in [15, 16, 19, 20].

The original method [15] actually comprises 2-stages of Preview Control (Fig. 4), where the 2nd stage acts as a "Dynamics Filter" [21,22] to correct for the expected error as a result of using the simplified ZMP equation. In comparison, toy-sized humanoids [23–25] use only the 1st stage of ZMP Preview based on the dynamics of the cart-table model. In one variation, [26] proposes a single-stage "General" ZMP Preview incorporating angular momentum with the cart-table dynamics.

ZMP Preview [15] has inspired an entire class of WPG that fix the footstep positions, with Wieber's method [16] putting explicit bounds on the trajectory of the ZMP for robustness, optimizations for speed of computation by Dimitrov [27], and the ability to quickly change footstep positions online [28, 29]. However, these algorithms are typically tested in simulation only, where it is much easier to incorporate the current state of the robot as feedback than it is on a real robot with noisy sensor measurements. We decided to test the original ZMP Preview algorithm both because it uses a full model of the robot as a Dynamics Filter, and because of its popularity, to determine how effective the method is on a full-sized robot. A quick overview of the algorithm is provided here.

Using the 3D-LIPM to model the robot dynamics, the COP (Center of Pressure) in the forward direction is:

$$p_x = x - \frac{z_c}{g}\ddot{x} \tag{7}$$

for which the inputs are the forward position of the COM x, acceleration of the COM $\ddot{x}$, constant height of the COM z_c, and constant gravity g. However ZMP Preview solves the inverse problem for which the COP is an input. Note the COP equations are decoupled and so the equation for the lateral direction p_y is similar [15].

As with the Polynomial WPG in Section II-A, the trajectories for the COP and COM are described by 3rd order polynomials.

A discretized system with constant sampling period T is developed [15] such that:

$$\hat{x}(k+1) = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \hat{x}(k) + \begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix} \dddot{x}(k) \tag{8}$$

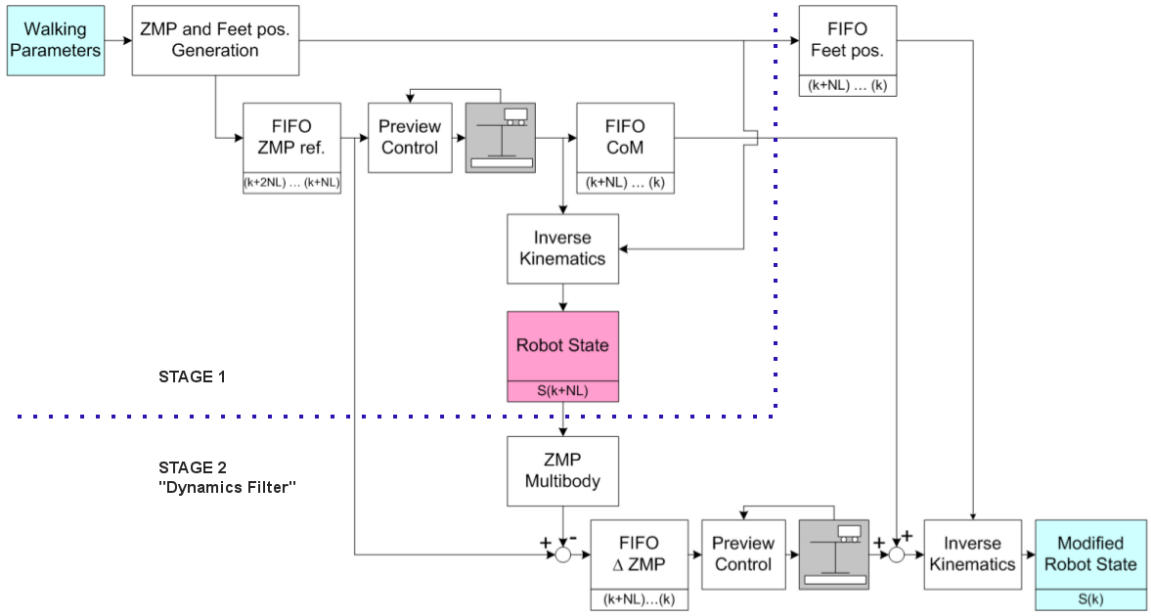

Fig. 4. WPG using 2-stage ZMP Preview [19]

where the state of the system is:

$$\hat{x}(k) = \begin{bmatrix} x(kT) \\ \dot{x}(kT) \\ \ddot{x}(kT) \end{bmatrix} \text{ for } k = 1, 2, ... \tag{9}$$

and the system input is the time-derivative of the acceleration of the COM (the jerk):

$$\dddot{x} = \frac{d}{dt}\ddot{x} \tag{10}$$

The goal is that each footstep is constrained to a certain position on the ground and ideally the COP is located at the center of the foot. ZMP Preview attempts to track this desired COP by minimizing the jerk:

$$\min_{\dddot{x}(k),\dddot{x}(k+1),...} \sum_{i=k}^{\infty} \frac{1}{2} Q \left(p_x(i+1) - p_x^{ref}(i+1)\right)^2 + \frac{1}{2} R \dddot{x}^2(i) \tag{11}$$

with constant weights Q and R.

At each step, the resultant COM position is used as the position of the pelvis center. Using IK (Inverse Kinematics) the current state and expected ZMP of a rigid multi-body model of the robot is computed. This expected ZMP is used to offset the error from the simple Cart-Table model of the robot used in the Preview Control formulation, and fed into a 2nd Preview stage (Fig. 4) which results in a more stable walking pattern.

III. Model Predictive Control

This section will briefly describe Wieber's method [16] that extends ZMP Preview to have explicit bounds on the trajectory of the ZMP, that should be more robust against external disturbances.

Wieber proposed solving the Quadratic Program for ZMP Preview (11) on a finite time horizon using Model Predictive Control:

$$\min_{\dddot{x}(k),...,\dddot{x}(k+N)} \sum_{i=k}^{k+N-1} \frac{1}{2} Q \left(p_x(i+1) - p_x^{ref}(i+1)\right)^2 + \frac{1}{2} R \dddot{x}^2(i) \tag{12}$$

with constraints on the ZMP:

$$\min_{\dddot{x}(k),...,\dddot{x}(k+N)} \frac{1}{2} \dddot{x}^2(k) \tag{13}$$

$$z_{k(min)} \leq z_k \leq z_{k(max)}$$

which ensures that the controller will keep the COP within a small, safe region of the support polygon, even when the walking pattern is disturbed by an unexpected external force.

This QP is solved analytically using an off-the-shelf solver. Given that a solution to the QP (13) must be computed at each time step, a WPG using this Model Predictive method will need sufficient computing power to solve the QP (13) at each time step, however [28] demonstrated a specialized solver with significantly faster speed.

IV. Hardware and Simulation

The platform used for this experiment is the HUBO2+ light-weight humanoid robot, which has 38 DOF and weighs 45 kg [18]. Position commands are sent to each joint at 200Hz from the controller running on Lubuntu Linux with the Xenomai framework for hard real-time performance, and the walking algorithms are implemented using the libRainbow and JRL Dynamics software libraries. Smooth walking is achieved by applying damping control at the

ankles and vibration compensation of the swing leg during SSP (Single Support Phase), ZMP control in DSP (Double Support Phase), and a landing timing controller adjusts the step period.

A dynamic simulation was created using Webots [30] and ODE (Open Dynamics Engine) [31]. The simulated model of HUBO2 (Fig. 2) consists of kinematic and inertial properties obtained from CAD data. The critical dimensions of the lower-body are shown in Fig. 2 and each foot is 0.22 x $0.15m$. The real robot has urethane bushings in each ankle to provide compliance, so this is modeled by a non-actuated ball joint with adjusted spring constant k.

In an upright straight-legged pose the height of the COM is $COM_z = 0.645m$. When the robot moves to a stable bent-knee pose the $COM_z = 0.6146m$, for joint angles:

$$\begin{aligned} \theta_{hip\ pitch} &= -21.8^\circ \\ \theta_{knee} &= \ \ 43.6^\circ \\ \theta_{ankle\ pitch} &= -21.8^\circ \end{aligned}$$

On the simulated robot, single-axis force sensors are positioned on the corners of the feet, which allows computation of the COP (Center of Pressure):

$$x_{cop} = \frac{\sum_{i=1}^{8}(F_{zi}p_{xi})}{\sum_{i=1}^{8}F_{zi}} \tag{14}$$

$$y_{cop} = \frac{\sum_{i=1}^{8}(F_{zi}p_{yi})}{\sum_{i=1}^{8}F_{zi}} \tag{15}$$

where p_{xi} and p_{yi} are the distance of the i^{th} force sensor from the respective ankle.

The real robot has a force-torque sensor in each ankle, between the urethane bushing and the foot, which measures the downward force F_z and moments M_x and M_y which are used to compute the ZMP (Zero Moment Point). Webots does not have a specific sensor for measuring torque, so a dummy joint was added to the simulation model and an ODE physics plugin is used to extract the forces and torques that satisfy the joint constraint during each iteration of the simulation.

Using the force-torque data and assuming a simple Cart-Table model [15] for the robot, the ZMP in the x and y direction is:

$$p_x = \frac{-\mathcal{T}_y}{F_z} \tag{16}$$

$$p_y = \frac{\mathcal{T}_x}{F_z} \tag{17}$$

where $\mathcal{T}_x$ and $\mathcal{T}_y$ are the ankle torques, and F_z is the force acting in the vertical direction. The ZMP is equivalent to the COP when the robot is stable and on a flat surface [32].

At each time step the COM (Center of Mass) for each link is used to calculate the robot's overall COM:

$$r_G = \frac{\sum_{i=1}^{40} r_i m_i}{\sum_{i}^{40} m_i} \tag{18}$$

where r_i is the position and m_i is the mass of the i^{th} link.

To find the Support Polygon, the convex hull of the contact points between the feet and the floor is calculated using the Qhull library.

A simulated environment of an indoor scene with various obstacles was constructed (Fig. 5).

V. EXPERIMENTS

A. SIMULATION

The three methods were first tested by making the simulated robot walk 6 steps in a straight line, to compare their performance with the expected results. The existing Parametrized Polynomial WPG has poor stability during yawing motions, so a more complex footstep path was tested where the robot encounters an obstacle and must make turns to avoid it.

B. REAL ROBOT

Each algorithm was used to generate a short 4 step pattern on the real robot, although it is difficult to isolate the relative performance of each WPG because the walking is very unstable without feedback control. The relative stability of each method is compared in the following section.

VI. RESULTS AND DISCUSSION

The walking patterns generated by the simulated robot avoiding an obstacle are shown in Fig. 7 – 9, using the Parametrized Polynomial, Preview Control and MPC algorithms. The actual COP (Center Of Pressure) and COM (Center of Mass) of the simulated robot are overlaid on the footstep pattern. The standard ZMP stability margin is considered, which is the distance from the COP to the edge of the SP (Support Polygon), thus for maximum stability the COP should be at the center of the SP.

The existing method of Parametrized Polynomial is unstable during turning on the real robot, and this is confirmed in simulation where the walk is stable but the COP is very close to the SP during turns (Fig. 7). Both Preview Control and MPC result in a higher stability margin for a simulated rigid-body humanoid on a flat surface, which was expected as the WPG incorporates the dynamics of a rigid multi-body model of the robot. In both cases there is some lateral and forward pitching of the COP which is due to inaccuracies in how Webots/ODE models the solid contact between the feet and floor. The MPC method Fig. 9 has a slightly better stability margin because the algorithm incorporates strict constraints on the ZMP margin.

However, testing the WPG (Walking Pattern Generators) on the real robot shows that the results in the ideal simulation world do not easily scale-up to a full-sized humanoid robot. A video showing initial testing of both WPG is available online [33]. Unmodeled compliance in the ankles, feet and floor causes instability problems for both pattern generators.

The HUBO2 robot has a compliant joint at the ankle created by a polyurethane bushing that is used to absorb impact forces during foot landing. The amount of ankle compliance can be varied by tightening a hex bolt, but it is tricky to get both ankles exactly the same. There is also

Fig. 5. Simulated HUBO2 robot walking around obstacles

compliance in the feet, due to the 4 rubber pads located on the underside, consisting of dual layers of hard and soft cell anti-slip rubber. Additionally, there is compliance in the floor where the tests were conducted, which is a suspended office floor typically found in high-rise buildings and demonstrative of the sort of non-ideal conditions in which future robotic assistants will need to operate. All this compliance is not accounted for in the original simulation model and the result is that the robot's pelvis tilts over and the swing foot lands on the floor too early in the walking phase. From the video [33], it can be seen that the short set of footsteps produced by the Parametrized Polynomial method appears to be slightly more stable, which is because the gait parameters used for the simulated robot were experimentally chosen from testing on the real robot with the stabilizer enabled. This means that the gait parameters for the Parametrized Polynomial WPG actually cancel the unmodeled compliance, whereas the Preview Control and MPC methods are predicting the behavior of an inaccurate rigid-body model.

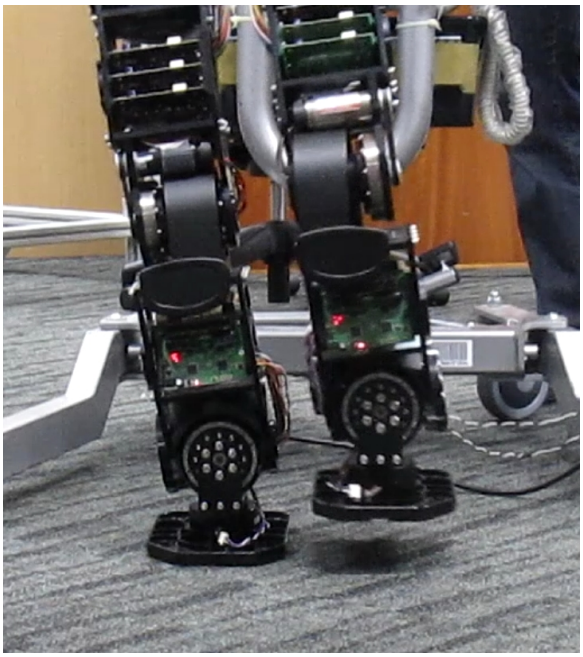

Fig. 6. Testing the Walking Pattern Generators on the real robot

To more accurately simulate the real robot, the spring constant in the ankles was set to $k = 2000$ and the response of the simulated HUBO was observed to more closely matche the actual robot. We plan to properly validate the simulation model by comparing data from the FT and IMU sensors on the real robot.

We have confirmed that the 2-stage ZMP Preview algorithm does produce good results on ideal terrain, but where there is significant unmodeled compliance in the robot and floor the Dynamics Filter incorporating the rigid multi-body model is not sufficient to compensate for the instability. We plan to modify the multi-body Dynamics Filter to model the springs in the ankles and incorporate an observer for feedback control.

Solving DAREs (Discrete-time Algebraic Riccati Equations) and QP (Quadratic Programs) can be computationally expensive, particularly for a low-power single-core CPU like that used in HUBO. The robot uses a control loop time of 5ms and the existing WPG using Parametrized Polynomials executes within a fraction of this time period. Our implementation of ZMP Preview uses RNEA (Recursive Newton Euler Algorithm) to compute the multi-body ZMP used in the 2nd Preview stage which requires significantly more computation time. On a Core i7 1.2GHz each iteration takes 11ms, and on an Atom N450 1.66GHz PC-104 Embedded PC each iteration takes 23ms. The compute time for each iteration of the MPC method is similar, as it also uses RNEA, but there is an initial delay of upto 4 min which we are investigating. It may be possible to double the computation speed by using the latest single-core i7 CPU, however at the time of writing such CPUs are not yet available for embedded PC platforms. If the Dynamics Filter incorporating an observer can produce a more stable walk, then we plan to optimize the dynamics

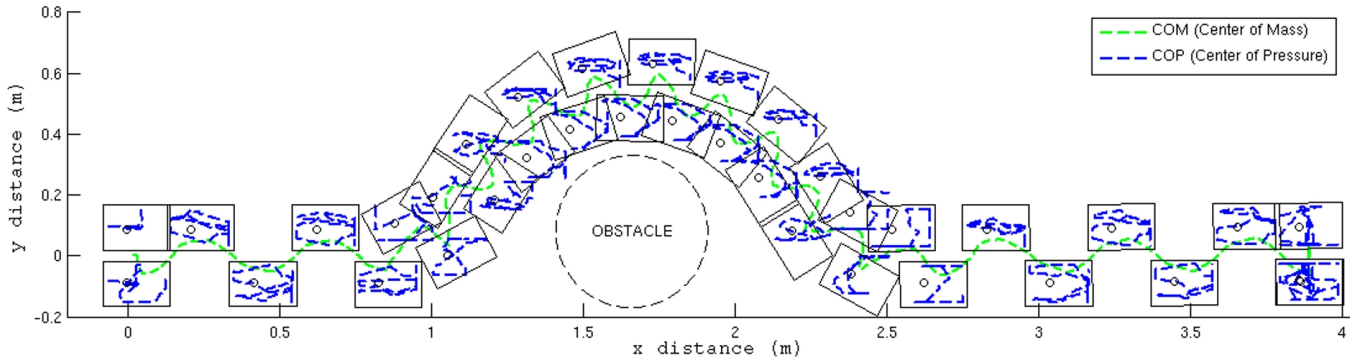

Fig. 7. Footstep path & evolution of COP whilst walking around obstacle, using Polynomial WPG in simulation

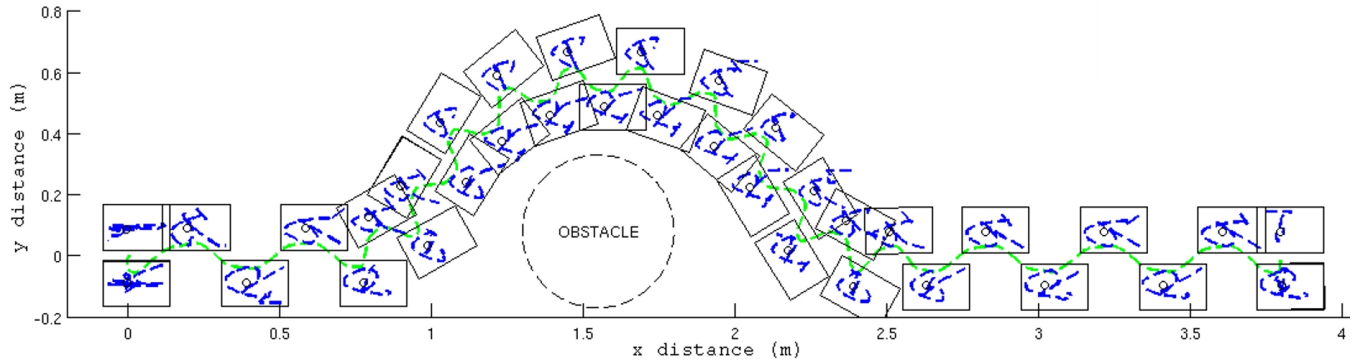

Fig. 8. Footstep path & evolution of COP whilst walking around obstacle, using ZMP Preview WPG in simulation

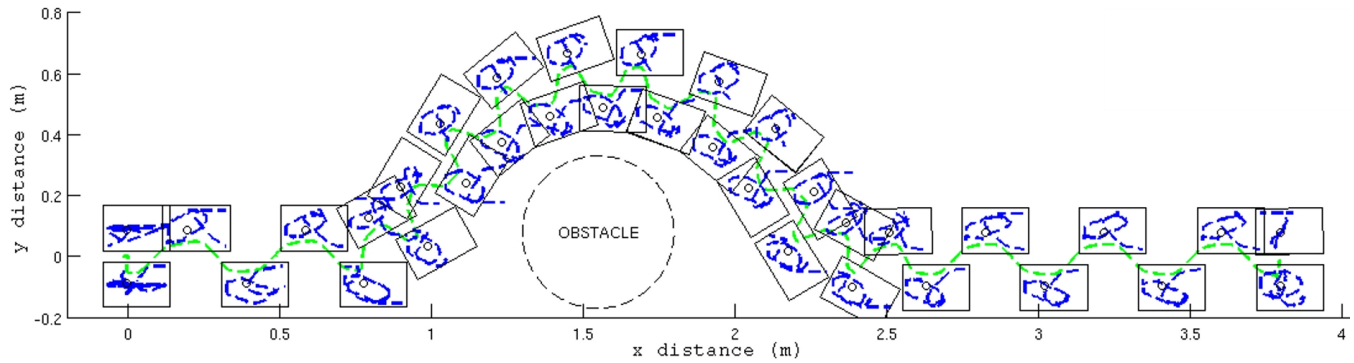

Fig. 9. Footstep path & evolution of COP whilst walking around obstacle, using MPC WPG in simulation

calculations.

Concerning observer-based feedback control, there are two interesting differences between previous implementations of ZMP Preview on the toy-sized and full-sized humanoids. The Aldebaran walk engine [24] uses the current position from joint angle sensors as feedback to the Preview Control, which is perhaps useful for such robots that exhibit a lot of 'play' in the joints. However this is not applicable to a larger humanoid utilizing high-gain position controllers that will accurately hold the target joint angles even whilst the robot is subject to large destabilizing torques. More complex feedback controllers are required to ensure that the feet make parallel contact with the the floor, and that the waist remains horizontal. Secondly, toy-sized robots such as NAO are inherently more stable. For comparison, the lateral foot-to-height ratio of NAO is 1:3.58, compared to HUBO2 which is 1:6.59 and thus more top-heavy.

So what is to be gained by complex model predictive algorithms such as this? The main benefit is being able to generate a COM trajectory from the footstep positions. But is it important to use the 2nd preview control stage incorporating the full dynamic model of the robot? Implementations of ZMP Preview for the toy-sized robot NAO only use a single stage [23, 24], combined with feedback of the actual robot state. From initial testing we have shown that the 2-stage ZMP Preview algorithm does not guarantee a more stable walking pattern on a full-sized humanoid where there is unmodeled compliance. With further experiments we hope

to determine if a MPC incorporating a more accurate non-rigid model of the robot could yield better results than simply parameter tweaking in the Parametrized Polynomial method.

The initial tests on the robot suggest that landing control of the swing foot has a bigger impact on stability than ZMP tracking and so we propose two potential approaches for developing a Walking Pattern Generator. Firstly is to develop a much more complex system that models the compliance in the robot and uses sensing to model the environment. It is easy to imagine that such methods would become inordinately complex as the robot is used in increasingly complex environments. Secondly is to use a rudimentary pattern generator based on simple LIPM (Linear Inverted Pendulum Model) combined with impedance control for the lower body to compensate for uneven or compliant ground surfaces.

VII. CONCLUSIONS AND FUTURE WORK

We have shown that when the Walking Pattern Generator incorporating a Dynamics Filter is applied to a full-sized humanoid robot, unmodeled compliance in the robot and the environment can negate the effectiveness of this approach. This leads to the question, if the robot or environment has significant compliance, is it better to use predominantly a predictive approach like Preview Control or a reactive approach consisting of a hierarchy of feedback controllers? As computing power increases, and humanoid robots are becoming more affordable, it is worth revisiting which WPG are actually applicable to real robots and hopefully we can establish some best-practice techniques in this area.

ACKNOWLEDGMENTS

The HUBO series humanoid robot platform was developed by Prof Jun-Ho Oh at the Humanoid Robot Research Center at KAIST (Korea Institute of Advanced Science & Technology) and is manufactured by Rainbow Inc.

Thank you to Dr Jungho Lee from Rainbow Co. for his expert knowledge of HUBO software, and to Dr Olivier Stasse from LAAS-CNRS for his assistance with the JRL humanoid dynamics software.

REFERENCES

[1] A. H. Adiwahono, B. Liu, N. Chen, T. W. Chang, A. D. Tran, X. Wu, and B. S. Han, "System Architecture for an Interactive Patrolling Humanoid Robot," in *Proceedings of the Workshop at SIGGRAPH Asia*, ser. WASA '12. New York, NY, USA: ACM, 2012, pp. 119–123.

[2] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The Intelligent ASIMO: System Overview and Integration," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2002, pp. 2478–2483 vol.3.

[3] Y. Ogura, H. Aikawa, K. Shimomura, A. Morishima, H. ok Lim, and A. Takanishi, "Development of a new Humanoid Robot WABIAN-2," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 76–81.

[4] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi, "Humanoid Robot HRP-2," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2004, pp. 1083–1090 Vol.2.

[5] B. Cho, J. Kim, and J. Oh, "Online Balance Controllers for a Hopping and Running Humanoid Robot," *Advanced Robotics*, vol. 25, no. 9-10, pp. 1209–1225, 2011.

[6] M. Vukobratovic and B. Borovac, "Zero-Moment Point - Thirty Five Years of its Life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2005.

[7] S. Yun and A. Goswami, "Momentum-based Reactive Stepping Controller on Level and Non-level Ground for Humanoid Robot Push Recovery," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 3943–3950.

[8] J. Englsberger, C. Ott, M. Roa, A. Albu-Schaffer, and G. Hirzinger, "Bipedal walking control based on Capture Point dynamics," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 4420–4427.

[9] I. Park, J. Kim, J. Lee, and J. Oh, "Online Free Walking Trajectory Generation for Biped Humanoid Robot KHR-3 (HUBO)," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 1231–1236.

[10] I. Park, J. Kim, and J. Oh, "Online Walking Pattern Generation and its Application to a Biped Humanoid Robot KHR-3 (HUBO)," *Advanced Robotics*, vol. 22, no. 2-3, pp. 159–190, 2008.

[11] F. Xue, X. Chen, J. Liu, and D. Nardi, "Real Time Biped Walking Gait Pattern Generator for a Real Robot," in *RoboCup 2011: Robot Soccer World Cup XV*, ser. Lecture Notes in Computer Science, T. Rfer, N. Mayer, J. Savage, and U. Saranl, Eds. Springer Berlin Heidelberg, 2012, vol. 7416, pp. 210–221.

[12] S. Hong, Y. Oh, Y.-H. Chang, and B.-J. You, "A Walking Pattern Generation Method for Humanoid Robots Using Least Square Method and Quartic Polynomial," *Humanoid Robots*, 2009.

[13] L. Righetti and A. Ijspeert, "Programmable Central Pattern Generators: an application to biped locomotion control," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 1585–1590.

[14] I. Park and J. Kim, "Fourier Series-based Walking Pattern Generation for a Biped Humanoid Robot," in *Proceedings of 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2010, pp. 461–467.

[15] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2003, pp. 1620–1626 vol.2.

[16] P. Wieber, "Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations," in *Proceedings of 6th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2006, pp. 137–142.

[17] A. Thant and A. Khaing, "Application of Cubic Spline Interpolation to Walking Patterns of Biped Robot," *World Academy of Science Engineering and Technology*, vol. 50, p. 2734, 2009.

[18] J. Heo, I. Lee, and J. Oh, "Development of Humanoid Robots in HUBO laboratory, KAIST," *Journal of the Robotics Society of Japan*, vol. 30, no. 4, pp. 367–371, 2012.

[19] O. Stasse, B. Verrelst, A. Davison, N. Mansard, F. Saidi, B. Vanderborght, C. Esteves, and K. Yokoi, "Integrating Walking and Vision to increase Humanoid Autonomy," *International Journal of Humanoid Robotics, special issue on Cognitive Humanoid Robots*, vol. 5, no. 2, pp. 287–310, 2008.

[20] O. Stasse, B. Verrelst, P. Wieber, B. Vanderborght, P. Evrard, A. Kheddar, and K. Yokoi, "Modular Architecture for Humanoid Walking Pattern Prototyping and Experiments," *Advanced Robotics, Special Issue on Middleware for Robotics – Software and Hardware Module in Robotics System*, vol. 22, no. 6, pp. 589–611, 2008.

[21] S. Kajita, K. Harada, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, , and H. Hirukawa, "A Dynamics Filter for ZMP Compensation by Preview Control (in Japanese)," in *In Proceedings of the 21st Annual Conference of the Robotics Society in Japan*, 2003, p. page 3A18.

[22] S. Kajita, *Humanoid Robot*. Japan: Ohmsha (in Japanese), 2005, ISBN 4-274-20058-2.

[23] J. H. Strom, "Dynamically Balanced Omnidirectional Humanoid Robot Locomotion," Honors Thesis, Bowdoin College, 2009.

[24] D. Gouaillier, C. Collette, and C. Kilner, "Omni-directional Closed-loop Walk for NAO," in *Proceedings of 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2010, pp. 448–454.

[25] Y. Jun, R. Ellenberg, and P. Oh, "Realization of Miniature Humanoid for Obstacle Avoidance with Real-time ZMP Preview Control Used for Full-sized Humanoid," in *Proceedings of 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2010, pp. 46–51.

[26] J. Park and Y. Youm, "General ZMP Preview Control for Bipedal Walking," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 2682–2687.

[27] D. Dimitrov, P. B. Wieber, H. J. Ferreau, and M. Diehl, "On the Implementation of Model Predictive Control for On-line Walking Pattern Generation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 2685–2690.

[28] H. Diedam, D. Dimitrov, P. B. Wieber, K. Mombaur, and M. Diehl, "Online Walking Gait Generation with Adaptive Foot Positioning through Linear Model Predictive Control," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 1121–1126.

[29] A. Herdt, D. Holger, P. Wieber, D. Dimitrov, K. Mombaur, and D. Moritz, "Online Walking Motion Generation with Automatic Foot Step Placement," *Advanced Robotics*, vol. 24, pp. 719–737, 2010.

[30] O. Michel, "Webots: Professional Mobile Robot Simulation," *Journal of Advanced Robotics Systems*, vol. 1, no. 1, pp. 39–42, 2004.

[31] ODE (Open Dynamics Engine) , "Open-source Simulator for Rigid Body Dynamics." [Online]. Available: http://www.ode.org

[32] P. Sardain and G. Bessonnet, "Forces Acting on a Biped Robot. Center of Pressure–Zero Moment Point," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 34, no. 5, pp. 630–637, Sept. 2004. [Online]. Available: http://dx.doi.org/10.1109/TSMCA.2004.832811

[33] D. Butterworth. Video accompanying this paper. [Online]. Available: http://david.butterworth.org.au/ACRA2013